

ТЕХНОЛОГИЯ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ ВАРИАНТОВ ОРГАНИЗАЦИИ РЕСУРСОВ ЛОКАЛЬНЫХ ВЫЧИСЛИТЕЛЬНЫХ СЕТЕЙ ПРИ ЗАДАННОЙ СТРУКТУРЕ РАСПРЕДЕЛЕННОЙ СИСТЕМЫ ОБРАБОТКИ ДАННЫХ

Abstract: *The technology of using imitation models of variants of organizing local net resource when proposed approach uniting conception of the net graph with methods of computer imitation.*

Key words: *technology of imitation modeling, local computer nets, distributed data processing system.*

Анотація: *Розглядається технологія використання імітаційних моделей варіантів організації ресурсів локальної обчислювальної мережі при проектуванні структури системи розподіленої обробки даних.*

Ключові слова: *імітаційні моделі, локальні обчислювальні мережі, розподілена обробка даних.*

Аннотация: *Рассматривается технология использования имитационных моделей вариантов организации ресурсов локальной вычислительной сети при проектировании структуры распределенной обработки данных.*

Ключевые слова: *имитационные модели, локальные вычислительные сети, распределенная обработка данных.*

1. Введение

При проектном моделировании вычислительных систем (ВС) и локальных вычислительных сетей (ЛВС) зачастую используются аналитические модели [1, 2], которые позволяют прогнозировать поведение технологических процессов обработки информации. Однако в аналитических моделях трудно учесть наличие конкуренции заказов пользователей за ресурсы ВС или ЛВС. Поэтому на более поздних стадиях проектного моделирования таких систем исследователи вынуждены использовать имитационные модели (ИМ). Один из подходов к исследованию сложных систем, основанный на сочетании аналитических и имитационных методов исследований, рассмотрен в работе [2]. Примеры полностью имитационного подхода к исследованию динамики использования ресурсов ЛВС приведены в работах [3, 4]. Для случаев, когда вычислительный процесс (ВП) в ВС или ЛВС представляет собой непрерывную смену исполнения запросов пользователей на различных узлах ЛВС, используются вероятностные ИМ ВП и рабочей нагрузки (РН) на ЛВС [5, 6]. Модели, построенные на возможности их представления в виде полумарковского процесса формирования запросов пользователей, уже не могут достоверно отобразить динамику появления запросов со стороны РН на ресурсы ЛВС, которые расположены в различных узлах ЛВС. В данной статье предлагается методологический подход, объединяющий концепцию анализа реализации стохастического сетевого графа на основе методов компьютерной имитации. Целью работы является изложение технологии имитации вариантов ВП для тех случаев, когда имеется возможность представить состав и структуру взаимосвязей между компонентами ВП в виде вероятностного сетевого графика (ВСГР) и использовать принцип формализации распределенной обработки информации в ВС и ЛВС на основе сочетания аппарата сетевого планирования и процедур метода стохастических испытаний.

2. Формализация ВП и РН на ЛВС на основе аппарата сетевого планирования

Все запросы пользователей, поступающие на обслуживание в ВС или ЛВС, можно разделить на следующие типы:

1. Требования пакетной обработки на исследуемом s -ом узле ЛВС.
2. Запросы пользователей в режиме диалогового взаимодействия с ВП, обрабатываемые на исследуемом s -ом узле ЛВС.
3. Требования удаленной пакетной обработки информации на других узлах сети.
4. Запросы пользователей удаленного диалогового режима.
5. Однократные взаимодействия между пользователями разных узлов типа сообщений.
6. Многократные взаимодействия между пользователями, находящимися на различных узлах, которые сопровождаются передачей информации в обе стороны по сети.

В общем случае РН на узлах ЛВС представляет собой комбинацию различных типов запросов пользователей.

По характеру обслуживания запросов пользователей используются следующие типы ИМ ВП и РН на узлах ЛВС, отображающие:

1. Внутреннее обслуживание запросов РН первого и второго типов.
2. Внешнее обслуживание после пересылки запроса на другие узлы.
3. Смешанное обслуживание, когда запросы обрабатываются внутри узла, но при этом присутствует однократная или многократная пересылка по сети результатов выполнения операций.

В работе [7] последовательность запросов пользователей на ресурсы узлов ЛВС представлена иерархической системой полумарковских моделей. В этих ИМ ВП представляется последовательностью программных модулей (ΠM_{ij}) , реализующих запросы i -го типа на центральном процессоре j -го узла (CPU_j) длительностью τ_{CPij} и использующих ресурс жесткого диска (HDD_j) объемом V_{HDij} . Порядок реализации ΠM_{ij} определяется матрицей вероятностей их следования друг за другом $\|P_{ij}\|$. Последовательный характер смен ΠM_{ij} и вероятностная природа расхода ими ресурсов узла $(\tau_{CPij}$ и $V_{HDij})$, а также дискретный характер реализации ВП позволяли зачастую адекватно отображать процесс обработки информации внутри узла и оценивать расходы ресурсов узлов ЛВС в виде полумарковского процесса. Когда ΠM_{ij} выполняются на различных узлах, строго последовательный характер их следования в ЛВС не соблюдается. Некоторые ΠM_{ij} могут выполняться на различных узлах ЛВС одновременно. Кроме того, взаимосвязи ΠM_{ij} по информации в ряде случаев могут быть строго детерминированы и порядок выполнения ΠM_{ij} в таких ситуациях будет обусловлен структурой базы данных сети. Для отображения параллельно-последовательного характера использования ΠM_{ij} ресурсов различных узлов ЛВС предложено использовать ВСГР, которые позволяют адекватно отобразить динамику распределенного использования ресурсов CPU_j и HDD_j . Аналогами ΠM_{ij} и узлов

(UZ_j) полумарковской модели (ПММ) в ВСГР являются микротехнологические операции ($MTXO_{ij}$) и события (SOB_j). Выполнение $MTXO_{ij}$ обеспечивается расходом:

- ресурса CPU_j (τ_{CPij}), обрабатывающего информацию с производительностью (ϑ_{CPj});
- ресурса HDD_j (V_{HDij}), обрабатывающего информацию с производительностью (ϑ_{HDj});
- стоимости реализации операции (c_{ij}).

В итоге основным параметром $MTXO_{ij}$ является время ее выполнения (T_{ij}), а дополнительными параметрами становятся c_{ij} , τ_{CPij} , V_{HDij} . Параметрами ресурсов s -го узла ЛВС являются их производительности CPU_s и HDD_s (ϑ_{CPs} , ϑ_{HDs}) и общий размер хранимой на HDD информации ($V_{\Sigma s}$). Этот общий ресурс каждого узла ЛВС одновременно используется многими $MTXO_{ij}$. Предлагаются известные регрессионные зависимости для определения времени выполнения $MTXO_{ij}$ (T_{ij}) по значениям параметров $MTXO_{ij}$ (τ_{CPij} , V_{HDij}) и производительности оборудования ЛВС:

$$T_{ij} = \varphi_1(\tau_{CPij}, \vartheta_{CPs}); T_{ij} = \varphi_2(\tau_{CPij}, \vartheta_{CPs}). \quad (1)$$

В отличие от классического определения сетевого графика каждая $MTXO_{ij}$ в $BCGP_k$, кроме времени ее выполнения, обладает набором следующих параметров (T_{ij} , τ_{CPij} , V_{HDij} , c_{ij}). При этом все эти параметры представляют собой случайные величины. Поэтому при описании $MTXO_{ij}$ в $BCGP_k$ ее параметры задаются соответствующими функциями распределения:

$$F_{1ij}(T), F_{2ij}(\tau_{cp}), F_{3ij}(V_{HD}), F_{4ij}(c). \quad (2)$$

В соответствии с классическим определением сетевого графика для события SOB_i вычисляются следующие статистики их реализации: разные и поздние сроки их свершения (t_{pi} и t_{ni}), резерв свершения события $R_i = t_{ni} - t_{pi}$. Независимые друг от друга $MTXO_{ij}$ выполняются параллельно на разных узлах ЛВС, а зависимые друг от друга $MTXO_{ij}$ инициируются только при свершении SOB_i в моменты их запуска на имитацию (t_{pi}). Все особенности информационного взаимодействия и порядок инициализации $MTXO_{ij}$ определяются структурой $BCGP_k$. В тех случаях, когда параметры $MTXO_{ij}$ являются постоянными величинами, $BCGP$ вырождается в обычный сетевой график, для анализа которого разработаны классические методики [7]. Целью этих методик является определение критического пути КРП, нахождение резервов свершения SOB_i (R_i), вычисление коэффициентов напряженности (Kn_{ij}) и коэффициентов резерва (Kp_{ij}) $MTXO_{ij}$ на сетевом графике. Под критическим путем в сетевом графике понимается

последовательность $\{MTXO_{ij}\}$, выполняемых на ресурсах узла ЛВС, у которых резервы предыдущих и последующих событий (SOB_i и SOB_j) равны нулю. Как известно [7], КРП определяет общее время выполнения запросов пользователей ЛВС (Tkr_k) в k -ом варианте запроса пользователями ресурсов ЛВС.

В случае анализа статистик реализации обычного сетевого графика по известным методикам [7] рассчитываются статистики реализации $MTXO_{ij}$: раннее начало (t_{pmij}), позднее начало (t_{lmiij}), раннее окончание (t_{poj}), позднее окончание (t_{loij}). В тех случаях, когда характеристики выполнения $MTXO_{ij}$ являются случайными величинами, аппарат сетевого планирования [7] не позволяет определить сроки свершения событий SOB_i и резервы их выполнения (t_{pi}, t_{li}, R_i). Такая ситуация имеет место в тех случаях, когда $MTXO_{ij}$ реализуются на разных узлах ЛВС и сами значения параметров $MTXO_{ij}$ необходимо задавать с помощью соответствующих распределений их значений (2). Отметим, что на практике постоянство структуры $BCGP_k$ и параметров $\{MTXO_{ij}\}$ является редким исключением. Поэтому вероятностный характер $BCGP_k$ и параметров $\{MTXO_{ij}\}$ обуславливает необходимость постановки имитационных экспериментов (ИЭ) с использованием процедур Монте-Карло [8].

3. Методика расчета и анализа параметров ВСГР на основе процедур Монте-Карло

Для решения проблем исследования вероятностных технологических процессов производства с помощью ИМ был разработан программно-технологический комплекс имитации (ПТКИ ТПП) [9], применение которого реализуется следующей последовательностью этапов.

Этап 1. Запись параметров $MTXO_{ij}$, входящих в ВСГР k -го типа запросов РН на ЛВС в информационную базу данных ПТКИ ТПП. При этом преобразуются описания $MTXO_{ij}$ во внутреннее представление. Взаимодействие этого комплекса с пользователем осуществляется на основе набора «меню» возможностей в режиме вопрос-ответ.

Этап 2. Расчет и анализ параметров $BCGP_k$ по методу Монте-Карло реализуется следующей последовательностью шагов.

Шаг 1. На l -ой реализации $BCGP_k$ ($s=1, \dots, N$) разыгрываются все значения параметров $MTXO_{ij}$ ($T_{ijl}, \tau_{CPijl}, V_{HDijl}, c_{ijl}$) с помощью соответствующих функций распределения (2).

$BCGP_k$ заменяется обычным сетевым графиком с постоянными значениями параметров $MTXO_{ij}$, к которому можно применить известную методику расчета и анализа статистик реализации операций $\{MTXO_{ij}\}$ и свершения событий $\{SOB_i\}$ [7].

Шаг 2. Осуществляется расчет ранних сроков свершения событий (t_{pi}). Формируются статистики расходов ресурсов ЛВС (V_{HDijl}), (τ_{CPijl}) и стоимости выполнения (c_{ijl}) при реализации

$\{MTXO_{ij}\}$. Вычисляются поздние сроки свершения событий (t_{nil}) по известному алгоритму [7], начиная с завершающего события в обратном порядке и кончая исходным событием.

Шаг 3. Рассчитываются резервы свершения событий $\{R_{il}\}$ и типовые статистики выполнения $MTXO_{ij}$ при реализации сетевого графика ($t_{рnijl}$, $t_{пnijl}$, $t_{роijl}$, $t_{поijl}$). Завершается анализ l-ой реализации $BCGP_k$ нахождением критического пути ($KPII_{kl}$) при реализации k-го запроса РН на ЛВС.

Шаг 4. В результате имитации выполнения N реализаций $BCGP_k$ в базе данных комплекса формируются выборки: значений статистик реализации событий $\{t_{ril}$, t_{nil} , $R_{il}\}$, статистик реализации $MTXO_{ij}$ $\{t_{рnijl}$, $t_{пnijl}$, $t_{роijl}$, $t_{поijl}\}$, критических путей $\{KPII_{sl}\}$. Каждой l-ой реализации $BCGP_k$ в этих выборках соответствуют l-ые номера статистик реализации $\{SOB_i\}$, $\{MTXO_{ij}\}$, $\{KPII_{lk}\}$.

Этап 3. Оптимизация ВСГР по данным ИЭ реализуется следующей последовательностью шагов.

Шаг 1. Формирование по выборкам математических ожиданий (M_z) и выборочных дисперсий D_z . При желании можно по выборкам построить эмпирические функции распределений $F(Z)$. Здесь под Z понимаем обозначение перечисленных ранее статистик свершения $\{SOB_i\}$, выполнения $\{MTXO_{ij}\}$ и длины критических путей в $BCGP_k$ ($T_{крпк}$).

Шаг 2. Осуществляется анализ $KPII_k$, представляющего собой последовательность чередования $MTXO_{ij}$ и SOB_i с нулевыми резервами времени их свершения ($R_{is} = 0$). В общем случае для N реализаций $BCGP_k$ может существовать множество $\{KPII_k\}$, в котором только некоторые пары (SOB_i , $MTXO_{ij}$) различны, а остальные могут не отличаться друг от друга. Поэтому исследователю сообщается диапазон реализации SOB_i , возникающих в $BCGP$ в одно и то же модельное время t_0 при различных реализациях $\{KPII_k\}$ в $BCGP_k$.

Шаг 3. С помощью статистической обработки результатов расчета статистик реализации $\{MTXO_{ij}\}$, $\{SOB_i\}$, $\{KPII_k\}$ формируются граф критических путей ($GRKPII_k$) и множество оценок значений коэффициентов напряженности выполнения $\{MTXO_{ij}\}$ [7]. Составляется также список тех SOB_i , которые имеют большие значения оценок математического ожидания резервов времени их свершения (R_i). Из этого списка выбираются те $MTXO_{ij}$, которые фигурируют затем в качестве кандидатов на изъятие от них резервов ресурсов для последующего направления этих ресурсов в распоряжение $MTXO_{ij}$, находящихся на наиболее вероятном критическом пути с целью уменьшения времени его свершения.

Шаг 4. Если множество $\{KPII_k\}$ достаточно большое, то из него формируется граф критических путей $GRKPII_k$. Далее реализуется следующая итерация поиска наиболее

вероятного критического пути, когда вместо BCP_k исследуется уже сам $GRKPI_k$. После нескольких итераций число вероятных критических путей существенно сокращается, и далее исследователь на основе анализа содержимого ветвей оставшегося графа $GRKPI_k$ может определить, какая из них является наиболее вероятной.

Шаг 5. Информация, сформированная на каждом шаге этапа 3, хранится в базе данных комплекса и может по запросу исследователя выводиться на экран дисплея в любом составе. Наряду с изменением параметров модифицируемых $MTXO_{ij}$ осуществляется переход на выполнение этапа 1.

4. Методика имитации распределенной обработки информации в ЛВС

Для имитации распределенной обработки информации в ЛВС можно использовать программно-технологический комплекс имитации (ПТКИ), реализующий транзактный способ организации квазипараллелизма на ЭВМ [9]. ПТКИ реализован на базе системы моделирования MICIS [10] и, согласно ее положениям объекты моделирования komponуются из динамических элементов (транзактов) и статических элементов (устройств). Транзакты движутся по ИМ от одного устройства к другому, которые представляют собой обслуживающие приборы с управляющими входами [10]. Поэтому при имитации распределенной обработки информации транзакты представляют собой ИМ запросов пользователей, а устройства являются ИМ реальных ресурсов узлов ЛВС. Направления движения транзактов по ИМ должны отражать структуру сети и зависимости между $MTXO_{ij}$ согласно BCP_k . Особенностью ИМ ВП в ЛВС является представление функционирования CPU в виде имитаторов использования ресурсов центрального процессора s-го узла ЛВС на четырех стадиях реализации $MTXO_{ij}$: ввод запросов пользователей (IN_s) длительностью (t_{INS}); обслуживание и распределение запросов по ИМ операционной системой (OS_s) длительностью (t_{OSS}); собственно обработка информации ($CALK_s$) по заказам пользователя с использованием ресурса CPU длительностью (t_{CALKS}) и ресурса HDD_s объемом (V_{HDIjs}); вывод результатов счета пользователю или возврат транзакта на дообслуживание (OUT_s). Передача информации между узлами ЛВС имитируется устройствами (СПД_s) длительностью ($t_{СПДs}$). Транзакты в ИМ могут обслуживаться подмоделями узлов ЛВС (ИМ UZL_s) по различным траекториям, зависящим от типа транзактов. Длительности пересылки информации по ИМ СПД_s до начала моделирования задаются матрицей средних времен пересылки информации $||t_{СПДs}||$. Итак, с помощью ИМ UZL_s моделируется конкуренция запросов пользователей за ресурсы различных узлов ЛВС. Для отображения динамики взаимодействия $MTXO_{ij}$ при реализации их на различных узлах ЛВС необходимо построить ИМ BCP_k и организовать взаимодействие ее с ИМ UZL_s .

ИМ BCP_k в среде ПТКИ реализуется сочетанием двух типов имитаторов: имитаторы событий в ВСР ($ISOB_i$ и $ISOB_j$), имитаторы микротехнологических операций ($ИМТХO_{ij}$). Каждый имитатор событий на выходе может создавать информационные транзакты ($INTR_{ij}$),

содержащие в себе информацию о запросах ресурсов узлов ЛВС соответствующей $MTXO_{ij}$ в виде функций распределения (2). Состав параметров $ИМТХО_{ij}$ зависит от типа запроса пользователя ($type_k$). Имитаторы микротехнологических операций ($ИМТХО_{ij}$) по информации, содержащейся в $INTR_{ij}$, создают управляющие транзакты (UTR_{ij}), в которых содержится заказ (Z_{ijl}) на ресурсы узлов ЛВС, необходимых для выполнения $MTXO_{ij}$ при l -ой реализации BCP_k по методу Монте-Карло. В Z_{ijl} управляющего транзакта UTR_{ij} содержится следующая информация:

$$Z = (ij; s, t_{INs}, t_{OSs}, t_{CALCs}, V_{HDS}, t_{OUTs}, t_{CПДs}, type_s, k_{nep}),$$

где ij – номера $MTXO_{ij}$, соединяющих события SOB_i с SOB_j ;

s – номер узла ЛВС;

$t_{INs}, t_{OSs}, t_{CALCs}, V_{HDS}, t_{OUTs}, t_{CПДs}$ – время обслуживания транзактов в устройствах ресурсов s -х узлов ЛВС;

$type_s$ – тип транзактов (3, 4, 5);

k_{nep} – количество пересылок транзактов между узлами ЛВС при выполнении запросов на ресурсы ЛВС от $ИМТХО_{ij}$.

Все UTR_{ij} поступают на вход ИМ UZL соответственно номерам узлов (s), содержащимся в телах заказов (Z_{ijl}), на которых имитируется динамика расхода ресурсов ЛВС. После окончания обслуживания в ИМ UZL_s управляющие транзакты UTR_{ij} возвращаются на имитаторы микротехнологических операций $ИМТХО_{ij}$ и затем ими уничтожаются. При этом фиксируются времена жизни UTR_{ij} ($t_{жij}$) и статистики длительности выполнения $ИМТХО_{ij}$ (t_{MTij}) в l -ой реализации BCP_k .

Методика имитации распределенной обработки информации в ЛВС с помощью множества подмоделей {ИМ UZL_s} и {ИМ BCP_k } (s -го номера узлов ЛВС, k -го номера типов запросов пользователей) реализуется следующей последовательностью этапов реализации взаимодействия алгоритмов BCP_k и алгоритмов обработки информации, распределенной по UZL_s.

Этап 1. Имитация выполнения { $ИМТХО_{ij}$ } согласно алгоритмам BCP_k . При срабатывании спусковой функции в устройствах-имитаторах событий $ИСОБ_i$ (когда завершилась обработка информации в $ИМТХО_{ij}$ и $ИМТХО_h$, которые предшествуют в BCP_k этому событию) на их выходах генерируются новые информационные транзакты $TRIN_{ij}$, поступающие соответственно на $ИМТХО_{ij}$. Каждая $ИМТХО_{ij}$ формирует соответствующий управляющий транзакт UTR_{ij} . В результате первого этапа формируется множество { UTR_{ij} }, элементы которого

созданы соответствующими $ИМТХO_{ij}$ и которые поступают каждый на вход той ИМ UZL_k , номер которой указан в теле UTR_{ij} .

Этап 2. Осуществляется имитация расхода ресурсов узлов ЛВС в ИМ UZL_s множеством $\{UTR_{ij}\}$, созданным на этапе 1. Процесс обслуживания каждого UTR_{ij} на s -ом узле ЛВС имитируется в ИМ UZL_s следующей последовательностью шагов.

Шаг 1. Выбор UTR_{ij} из общей очереди ИМ UZL_s и имитация устройством-имитатором процесса ввода информации ($ИИN_s$) длительностью ($t_{ИN_s}$).

Шаг 2. По окончании этой операции UTR_{ij} поступает в очередь к устройству-имитатору функционирования операционной системы ($ИОS_s$) длительностью ($t_{ОS_s}$), где определяется направление движения UTR_{ij} по устройствам-имитаторам ресурсов узлов ЛВС.

Шаг 3. Имитируется расход ресурса CPU_s транзактами UTR_{ij} на устройстве-имитаторе $CALC_s$ длительностью ($t_{CALC_{sij}}$). По завершении этой операции UTR_{ij} либо возвращается на дообслуживание в очередь к устройству $ИИN_s$, либо направляется к устройству-имитатору $ИСПД_s$ на дообслуживание его на других узлах ЛВС, либо завершает свой цикл обслуживания на устройстве-имитаторе вывода информации $ИОUТ_s$ длительностью ($t_{ОUТ_s}$). Длительности операций пересылки UTR_{ij} через $ИСПД_s$ перед началом моделирования задаются матрицей средних времен пересылок информации $\|t_{СПД_s}\|$. По завершении запланированного числа пересылок ($k_{Пij}$) транзакта UTR_{ij} внутри ИМ ЛВС имитируется возврат UTR_{ij} на $ИМТХO_{ij}$.

Этап 3. Имитация завершения l реализации $ВСГР_k$ осуществляется на устройстве-имитаторе завершающего события ($SOB\ POGLn_k$), где n_k – номер завершающегося события k -го запроса пользователя. Фиксируется статистика l -ой реализации $ВСГР_k$:

$$ST_{kl} = (T_{КРКl}, \{t_{pil}, t_{Пil}, R_{il}\}, \{t_{PHijl}, t_{ПHijl}, t_{poijl}, t_{Поijl}\}, КРП_{kl}),$$

где $T_{КРКl}$ – критическое время l -ой реализации $ВСГР_k$;

$\{t_{pil}, t_{Пil}, R_{il}\}$ – статистики свершения событий $i=1, \dots, n_k$ в $ВСГР_k$;

$\{t_{PHijl}, t_{ПHijl}, t_{poijl}, t_{Поijl}\}$ – определенные ранее статистики реализации $МТХO_{ij}$;

$КРП_{kl}$ – критический путь в l -ой реализации $ВСГР_k$.

Кроме формирования ST_{kl} контролируется момент окончания имитации реализаций $ВСГР_k$ ($l \leq N$) по методу Монте-Карло [7] и либо организуется повторение следующей реализации $ВСГР_l$ с модификацией числа итераций ($l:=l+1$), начиная с этапа 1, либо осуществляется переход к этапу 4.

Этап 4. Обработка результатов имитации N реализаций $ВСГР_k$.

К началу этого этапа уже накоплена статистика реализации событий, микротехнологических операций в виде соответствующих выборок статистик, сформирован граф критических путей {GRKPP} и требуется по этим выборкам либо вычислить математические ожидания и дисперсии статистик свершения событий и реализации микротехнологических операций, либо построить эмпирические функции их распределений. Кроме того, на этапе 4 по GRRHG определяются вероятные критические пути с оценкой вероятностей реализации каждого из этих путей согласно методике, изложенной в разд. 3.

5. Имитационная модель распределенной обработки информации в ЛВС

Простейший пример ИМ ЛВС, состоящей из одной ИМ ВСГР₁ и двух ИМ UZL_s ($s=1,2$), с распределенной обработкой информации приведен на рис. 1, ИМ ВСГР₁ состоит из 8 устройств-имитаторов событий ($ИСОБ_j$) $i=1,\dots,8$ и 11 устройств-имитаторов микротехнологических операций $ИМТХО_{ij}$. В этой ИМ отображен параллельно-последовательный характер выполнения $МТХО_{ij}$ и запуск $ИМТХО_{ij}$ при свершении событий $ИСОБ_i$. Каждая $ИМТХО_{ij}$ генерирует управляющие транзакты $УТR_{ij}$, поступающие во входные очереди ОЧ₁₁ и ОЧ₂₁ соответственно ИМ UZL₁ и ИМ UZL₂.

ИМ UZL_s состоят из четырех устройств-имитаторов ресурсов CPU ЛВС (ИИН_s, ИОС_s, ИСАЛС_s, ИОУТ_s) и имитатора устройства пересылки информации по ЛВС (СПД_s). На вход ИМ UZL_s поступают две группы транзактов: TUZL_s и $УТR_{ij}$. Первая группа транзактов формирует внутренние запросы узла ЛВС на ресурсы CPU_s. Такие запросы рождаются либо при диалоговом характере взаимодействия пользователей на узлах ЛВС, либо при пакетной обработке заказов пользователей внутри узла ЛВС. Остальные типы запросов пользователей могут использовать распределенную обработку информации в ЛВС, и динамика расхода ими ресурсов узлов ЛВС определяется в ИМ $ВСГР_k$. Первый тип транзактов TUZL_s формируется на входе ИМ UZL_s генератором (GEN TUZL_s), обслуживается наряду с $УТR_{ij}$ устройствами-имитаторами ресурсов CPU_s и затем либо уничтожается поглотителем (ПОГЛ_s), либо возвращается на повторное обслуживание в приоритетную очередь ОЧ_{3s} узла ЛВС. По завершении цикла обслуживания $УТR_{ij}$ устройство ПОГЛ_s возвращает $УТR_{ij}$ соответственно на $ИМТХО_{ij}$ в ИМ $ВСГР_k$. Как видно из рис. 1, учет расхода ресурсов CPU_s запросами первого типа позволяет отобразить конкуренцию запросов пользователей ЛВС за ресурсы ЛВС.

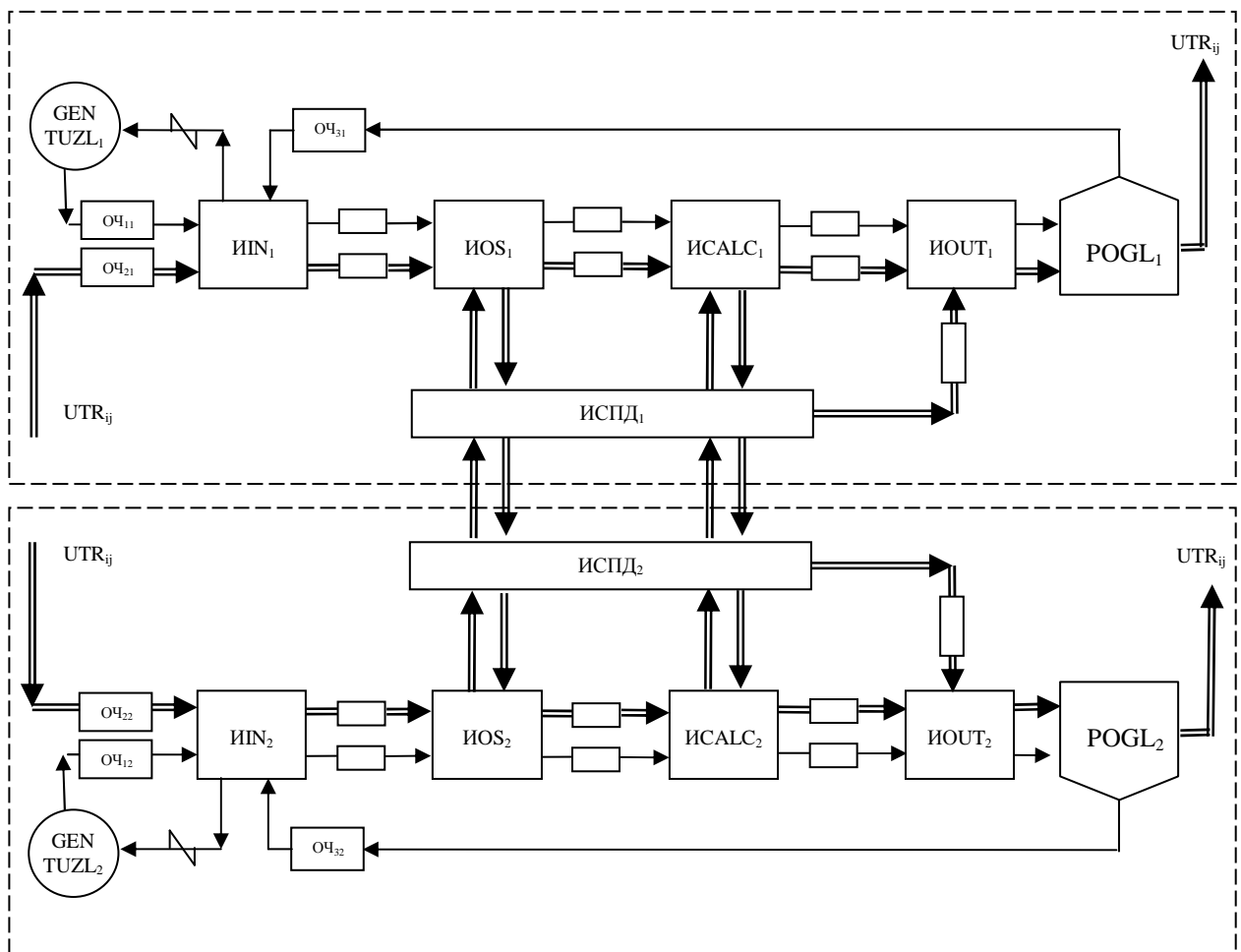
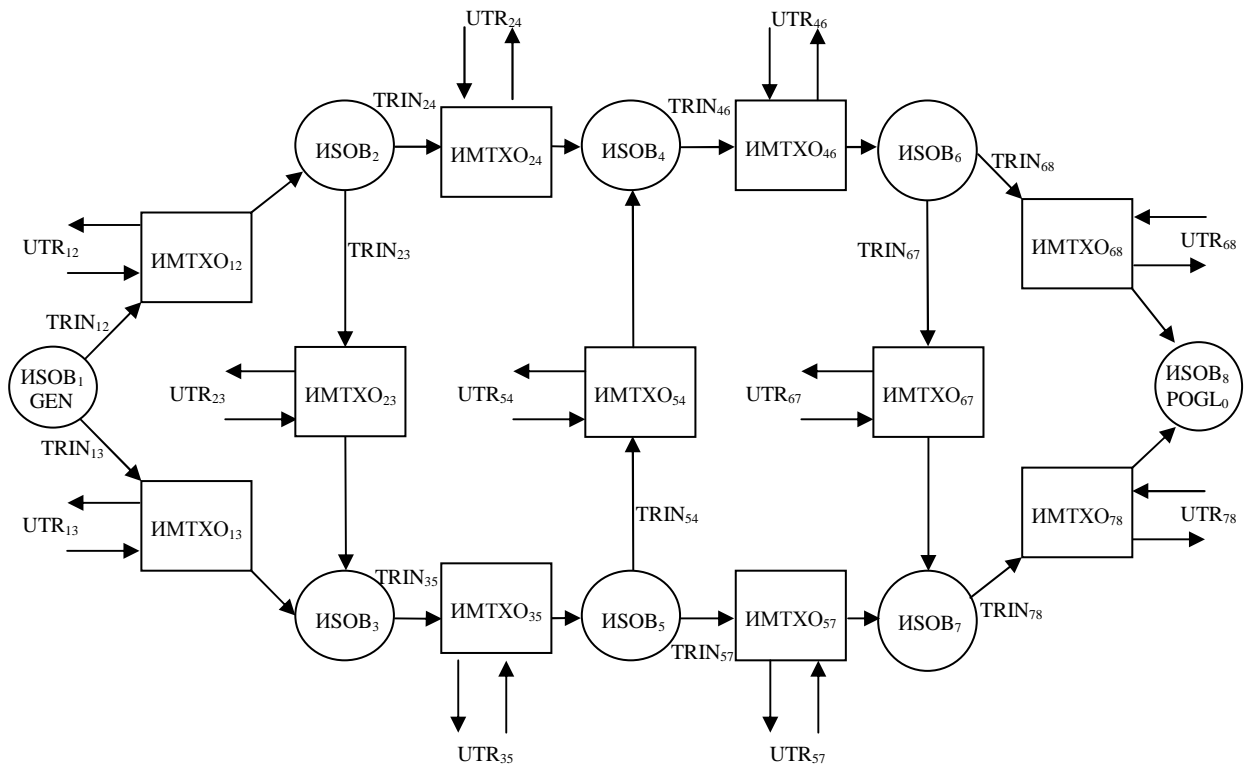


Рис. 1. Имитационная модель распределенной обработки информации в ЛВС взаимодействия пользователей на узлах ЛВС, либо при пакетной обработке заказов пользователей

6. Выводы

Основным результатом данной работы является иллюстрация нового подхода к исследованию вычислительного процесса для случая распределенной обработки в ЛВС. Дополнительными трудностями имитационного моделирования таких систем является необходимость учета следующих аспектов: параллельно-последовательный характер взаимодействия программных модулей, выполняемых одновременно по запросам пользователей на различных узлах ЛВС; необходимость учета конкуренции запросов пользователей за ресурсы узлов ЛВС; требования учета географии расположения информации, обрабатываемой по запросам пользователей на различных узлах ЛВС; влияние состава и топологии ЛВС на организацию вычислительного процесса в узлах сети. Для отображения этих разнородных требований при моделировании в сети предложены новые принципы формализации и методики организации имитации ВП в ЛВС, основанные на сочетании аппарата сетевого планирования, процедур метода статистических испытаний, расширения транзактно-процессного способа имитации за счет информационной подкраски транзактов и функциональной специализации имитаторов расхода ресурсов распределенных по узлам ЛВС. Основной трудностью в реализации такого сочетания являлась необходимость наложения структуры графа взаимосвязей между PM_{ij} , в совокупности отображающего модель РН на узлах ЛВС, на топологию связей ресурсов в сети и учета наличия запросов пользователей узлов ЛВС, которые в приоритетном диалоговом режиме активно используют ресурсы самих узлов ЛВС.

Большинство исследований ВП в ЛВС было основано на использовании полумарковских моделей. Но такой подход эффективен только в тех случаях, когда ЛВС уже существует, известен статистический портрет РН на узлах ЛВС и ставится задача имитационного моделирования для перераспределения ресурсов узлов ЛВС при их адаптации к требованиям РН. Предложенные в данной работе методики имитации распределенной обработки информации в ЛВС обладают преимуществом перед имитацией с помощью полумарковских моделей в тех случаях, когда реальная ЛВС еще не существует и решается задача выбора состава и размещения ресурсов узлов ЛВС под требования РН на узлах ЛВС. На простейшем примере ИМ ВП в ЛВС, приведенном на рис. 1, была проиллюстрирована возможность применения новой методики имитационного моделирования распределенной обработки информации в ЛВС.

СПИСОК ЛИТЕРАТУРЫ

1. Зайченко Е.Ю. Анализ структуры глобальных вычислительных сетей. – К.: ЗАО «Укрспецпроект», 1998. – С.108.
2. Коваленко И.Н., Николаев А.Н. Приближенный расчет и оптимизация надежности. – К.: Наукова думка, 1989. – 182 с.
3. Демиденко О.М., Максимей И.В., Агеенко И.В. и др. Имитационное моделирование вычислительного процесса в узлах локальной сети // УСиМ. – 2000. – №5/6. – С.101–107.
4. Демиденко О.М., Максимей И.В. Проектное моделирование вычислительного процесса в локальных вычислительных сетях. – Мн.: Белорусская наука, 2001. – 252 с.
5. Демиденко О.М., Максимей И.В. Влияние конкуренции запросов пользователей за ресурсы вычислительного процесса на организацию вычислительного процесса // Математические машины и системы. – 2001. – № 2. – С. 3–9.
6. Демиденко О.М. Технология мониторинга и адаптации вычислительного процесса под рабочую нагрузку на локальную вычислительную сеть. – Мн.: Белорусская наука, 2002. – 193 с.
7. Жогаль С.И., Максимей И.В. Задачи и модели исследования операций: Учебное пособие. – Ч. 1: Аналитические модели исследования операций. – Гомель: БелГут, 1999. – 103 с.

8. Максимей И.В., Серегина В.С. Задачи и модели исследования операций: Учебное пособие. – Ч. 2: Методы нелинейного и стохастического программирования. – Гомель: БелГут, 1999. – 103 с.
9. Максимей И.В., Левчук В.Д., Маслович С.Ф. и др. О проблемной модификации программно-технологического комплекса имитации технологических процессов производства // Известия Гомельского государственного университета им. Ф. Скорины. – 2003. – № 3(18). – С. 38–41.
10. Максимей И.В., Левчук В.Д., Жогаль С.П. Задачи и модели исследования операций: Учебное пособие. – Ч. 3: Технология имитации на ЭВМ и принятие решений. – Гомель: БелГут, 1999. – 150 с.